2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN)
31 July – 2 August 2025, Rangpur, Bangladesh

2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN) | 979-8-3315-9694-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/QPAIN66474.2025.11171775

Identification of AI Generated Texts using an Ensemble Method

Md. Siam Ansary[1], Nawshin Tabassum Tanny[2]
[1,2]Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
141 and 142, Love Road, Tejgaon, Dhaka-1208
[1]ORCiD: 0000-0002-4942-2541, [2]Email: tanny.cse@aust.edu

***Abstract*—Since the rise in popularity of artificial intelligence, it has found its way into many aspects of everyday life which we once thought to be quite impossible. Today, AI tools are commonly used to generate various types of text, including academic papers and business reports. Although AI has shown remarkable potential and capability in producing text, it is increasingly important to set apart between content written by humans and that generated by AI, especially to protect against misuse and fraud. In this study, we explored an ensemble learning approach using three different BERT-based embeddings, achieving results that scored almost perfectly on the evaluation metrics. The code used in our experiments will be publicly shared to allow others to verify and replicate our findings.**

***Index Terms*—ensemble learning, natural language processing, artificial intelligence, text categorization**

I. Introduction

In these latter decades past, manifold and earnest inquiries hath been undertaken across diverse realms of artificial intelligence. In this present day, the workings of AI are employed in manners erstwhile deemed unthinkable, a most notable instance being the crafting of text by means of cunning and advanced engines of artificial intelligence.

Today, a wide range of AI-powered platforms and tools such as Gemini, ChatGPT, Microsoft Copilot, and QuillBot are available for generating or paraphrasing text. While these tools have undoubtedly simplified many tasks, relying too heavily on them can pose serious risks. Increasingly, students are submitting academic reports produced entirely by AI, and professionals are using these tools to draft business documents with little to no personal input. This not only diminishes the originality and authenticity of the content but also opens the door to widespread misuse, such as the creation of fraudulent documents. Being able to detect AI-generated text is becoming crucial, especially in contexts where authenticity matters.

On a publicly available dataset, we have conducted our research. As a classification model, we have used an ensemble learning technique. In an ensemble learning method, various classifiers are combined for a final prediction. We believe ensemble method is the best way to tackle AI generated text detection as human written texts can have different writing styles based on who wrote them, and also, AI generated texts can be produced in many ways via various tools and platforms and each of them may adapt a different style. An ensemble method will be more appropriate to tackle different writing or generation styles and do a prediction. The implemented ensemble model consists of eight classifying models and they are Random Forest, Adaptive Boosting (AdaBoost), Categorical Boosting (CatBoost), Stochastic Gradient Descent (SGD), Light Gradient Boosting Machines (LGBM), Bootstrap Aggregating (Bagging), Extra Tree and Extreme Gradient Boosting (XGBoost). These eight models have been tuned for preparing the ensemble technique for achieving impeccable results.

In the following sections, we have discussed related prior experiments, our methodology and evaluation results.

II. Literature Review

We have studied the existing researches on AI generated text detection to gain insights which can be very helpful.

Shah and his fellows [1] did employ a hybrid manner of discerning texts begotten of artificial craft, drawing upon the style and form of the writing. They did take heed of many matters: the lexical traits, the ease of reading, and the breadth and depth of the tongue employed. Custom scrolls of text were prepared by these scholars for their inquiry. In their trials, they did first set to classify the writings by means of Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, and Gradient Boosting. Thereafter, the findings of the two most able classifiers were joined with the wisdom of two explainable artifices, SHAP and LIME by name. This blended endeavour yielded results most hopeful and worthy.

Zhou and Wang [2] did labour upon the discerning of texts begotten by artifices of intelligence across diverse domains. By the joining together of sundry datasets, a single corpus was fashioned for their study. The learned men employed RoBERTa-Ranker, a refined form of RoBERTa, wherein was set a margin ranking loss function, and thereto was added a mean pooling layer upon the layer of encoding.

Guo et al. [3] did labour upon a pipeline for the discerning of text wrought by AI, wherein they did entwine a multi-level contrastive learning craft with the art of multi-task learning. First did they pre-encode the training samples and draw forth their several features. Thereafter, for any given text, the likeness 'twixt the encoded features and each vector harboured within the store of feature vectors was reckoned. At the last, forsooth, they did employ the method of K-nearest neighbours to divine the final judgment.

McGovern et al. [4] opinionated that with appropriate features, it is possible to detect AI generated texts using simple classifiers. The researchers used a GradientBoost classifier along with three feature sets which were word n-grams, character n-grams and parts-of-speech n-grams. In the study a number of datasets were used and the proposed methodology showed promising results.

Abbas [5] worked on AI generated text identification utilising a zero shot prompt along with Sentence BERT (SBERT). Also, graph convolutional network model was integrated which enhanced the accuracy. The PAN-AP-2019 dataset had been used in the experiment.

The researchers in a study [6] worked with multiple ML and DL classification models for AI-generated text detection. They created a dataset with 509 descriptive question-answers where the answers were prepared by both humans and the Chat-GPT engine. In the study, the RoBERTa-based model outperformed others.

Alhijawi et al. [7] created a dataset of 3000 records for LLM generated content detection. The records were of three types - Human written, AI generated, Mixed and each type had 1000 records. The researchers experimented with various approaches and presented a hybrid model of MLP and CNN which achieved a highest accuracy of 0.876 in the experiment.

III. Methodology

A. The Dataset

We have used a publicly available dataset from Kaggle [8] uploaded by Darek Kłeczek [9] for AI Generated text detection. The dataset has 44868 text records and 27371 records are human written while the rest are AI generated. The AI generated texts are labeled as 1 while texts written by humans are labeled as 0. The dataset is available in CSV file format. This public dataset had been created in combination of multiple public datasets, documents and LLM generated texts.

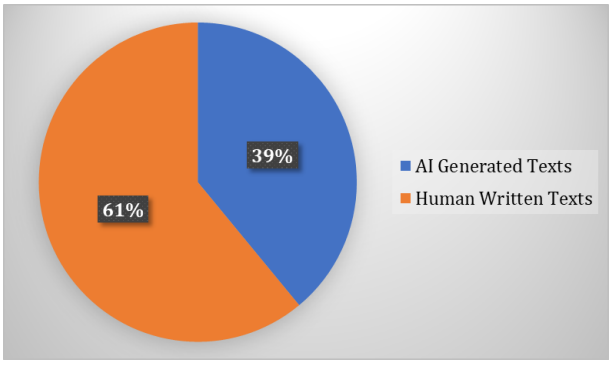

Fig. 1: Percentage of Category wise Records in Dataset

B. Application of the Ensemble Classification Method

Before applying the ensemble classifier, we have performed some pre-processing on the dataset. Firstly, the dataset has been checked for any duplicate values and if found, they were removed and the dataset was shuffled.

Later, the texts of the dataset were converted into numerical vectors embeddings. We have used BERT Based transformer models for the task.

In our experiment, we have done three set-ups. In the first set up, for embedding all-MiniLM-L6-v2 [11] is used; in the second set up, all-mpnet-base-v2 [12] is employed and in the last set up, we have used bert-base-uncased [13].

TABLE I: Different Set ups in the Experiment

Set Up	Classifier	Embedding Model
Set up 01	Proposed Ensemble Model	all-MiniLM-L6-v2
Set up 02		all-mpnet-base-v2
Set up 03		bert-base-uncased

all-MiniLM-L6-v2 is a small plus fleet-footed pre-trained model, a nimble transformer of light burden. This model was trained through the craft of contrastive learning, that it might yield sentence embeddings rich in meaning and sense — wherein the likeness of thought and intent may be rightly discerned.

all-mpnet-base-v2 is a more accurate and powerful model than all-MiniLM-L6-v2. It stems from the MPNet architecture with excellent performance.

bert-base-uncased is a pre-trained model that was forged as part of the first BERT architecture. It hath been trained upon a vast corpus of the English tongue and doth make use of uncased text — that is to say, it turneth all words to their lowercase form, casting aside all marks of upper and lower case, making no distinction between them.

After we get the sentense embeddings, they are then normalized into a consistent range. This has been done so that no particular dimension can add extra bias due to having larger values.

Dataset was portioned into seventy:thirty for the training and testing purposes.

In our implemented ensemble learning method, there are eight classifiers. They are as follows.

- Random Forest Classifier: Our implementation of Random Forest works with 300 trees having a max depth of 10 for each tree and a minimum of 4 samples to split an internal node which helps with increased robustness and reduced overfitting.
- AdaBoost Classifier: The implemented AdaBoost model works with 200 weak learners, each with moderate impact due to learning rate of 0.5. It used a fixed random seed of 42 for consistent results.
- CatBoost Classifier: The implemented CatBoost model is moderately complex and regularized. The classifier trains 500 trees with a slow learning rate of 0.03 limiting tree complexity with depth of 6 with L2 regularization to reduce overfitting.
- SGD Classifier: In the implementation, the SGD classifier holds a good balance of L1/L2 penalties with elastic net. It uses logistic regression as the loss function which is well suited for binary classification specifically. The

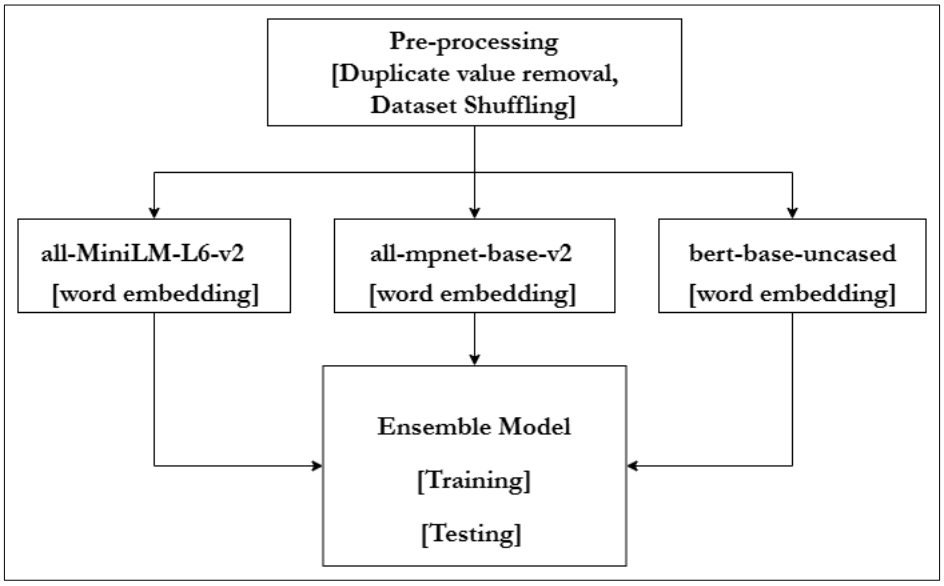

Fig. 2: Overview of Methodology

classifier holds alpha value of 1e-4 as regularization strength with 15% of L1 and 85% of L2. The model holds 2000 epochs for training with early stopping for preventing overfitting.

- LGBM Classifier: In our employed implementation, the classifier is highly regularized, which is ideal for controlling overfitting. The approach uses 1000 trees and a learning rate of 0.01 for gradual, robust learning. The complexity is balanced with maximum number of leaf nodes in a tree being 32, maximum depth of each tree being 8, minimum number of data points for a leaf node being 30 with 80% of data and 80% of feature being used for each tree which helps with increased speed, diversity and less overfitting. L1 and L2 regularization of 0.1 being applied help with generalization and reducing overfitting.
- Bagging Classifier: In our experiment, the Bagging classifier is set up with 100 base models, each of which is trained on 80% of the data and 80% of the features and, henceforth, the classifier is able to increase diversity and randomness along with reducing overfitting.
- Extra Tree Classifier: In our employed model, the classifier has 300 trees for improved accuracy with limiting depth of 10 for preventing overfitting, splits of 4 for avoiding overly small ones.
- XGBoost Classifier : In our implementation, the classifier has 500 estimators for potential better accuracy, 0.05 learning rate for better generalization, max death of each tree being 5 along with 0.8 of randomly sampled feature and training data of each tree for less overfitting, L1 regularization of 0.5, L2 regularization of 1 with log loss metric for evaluation during training.

The ensemble model is at first trained using the training chunk of dataset and then its performance is evaluated with the test segment dataset.

C. Evaluation Results

For evaluation of the text classification task, we have used Accuracy, Precision, Recall and F1 score metrics. The outcomes of our ensemble model is illustrated.

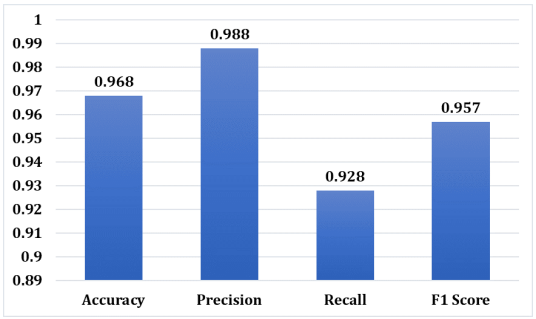

Fig. 3: Performance of the Proposed Ensemble Approach in the first Set up

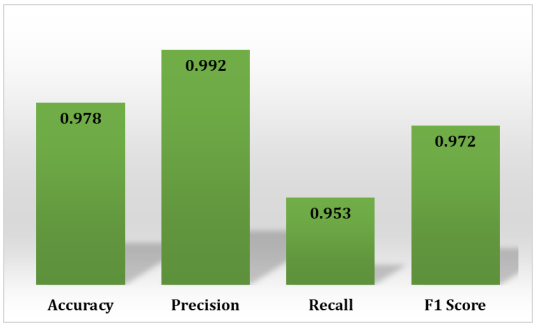

Fig. 4: Performance of the Proposed Ensemble Approach in the second Set up

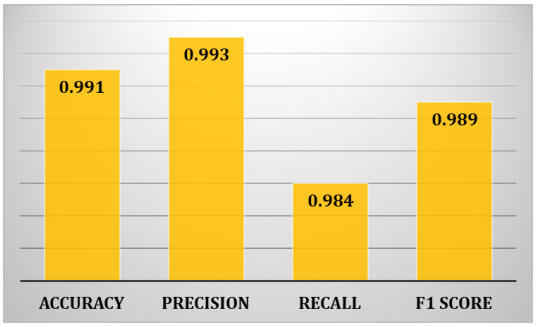

Fig. 5: Performance of the Proposed Ensemble Approach in the third Set up

We observe that the proposed ensemble method achieves excellent outcomes in all set ups but the best results come when for embedding, bert-base-uncased is used.

bert-base-uncased is trained with a masked language modeling objective and it is able to generate contextualized token-level embeddings. The model holds more depth as well as size and is better suited for finetuning.

On the other hand, the embedding models of set up one and set up two are trained using a contrastive learning objective and have less size and depth compared to bert-base-uncased.

IV. The Availability of the Resources

The resources related to the study will be publicly available on a GitHub repository `MdSiamAnsary/ClassificationTasks` [14].

V. Conclusion and Future Works

Identification of AGT is a climacteric task in today's life. Due to heavy usage of internet and AI tools, it is quite difficult to identify such texts. Rather than using standalone classifiers, we have used ensemble learning technique for the classification task and it has yielded excellent result with accuracy of 0.991. We aspire to create an application, whether mobile, desktop or web, on top of the classifier model, so that people can utilize this ensemble model for real life use. In future, we hope to perform the task on specialized texts. Also, we intend to explore possibilities and scopes for improving the evaluation outcome of the employed model.

References

[1] Aditya Shah, Prateek Ranka, Urmi Dedhia, Shruti Prasad, Siddhi Muni, and Kiran Bhowmick. "Detecting and unmasking AI-generated texts through explainable artificial intelligence using stylistic features." International Journal of Advanced Computer Science and Applications 14, no. 10 (2023).

[2] You Zhou, and Jie Wang. "Detecting AI-Generated Texts in Cross-Domains." In Proceedings of the ACM Symposium on Document Engineering 2024, pp. 1-4. 2024.

[3] Xun Guo, Yongxin He, Shan Zhang, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. "Detective: Detecting ai-generated text via multi-level contrastive learning." Advances in Neural Information Processing Systems 37 (2024): 88320-88347.

[4] Hope McGovern, Rickard Stureborg, Yoshi Suhara, and Dimitris Alikaniotis. "Your Large Language Models Are Leaving Fingerprints." GenAIDetect 2025 (2025): 85.

[5] Halah Mohammed Abbas. "A Novel Approach to Automated Detection of AI-Generated Text." Journal of Al-Qadisiyah for Computer Science and Mathematics 17, no. 1 (2025): 1-17.

[6] Maktabdar Oghaz, Mahdi, Lakshmi Babu Saheer, Kshipra Dhame, and Gayathri Singaram. "Detection and classification of ChatGPT-generated content using deep transformer models." Frontiers in Artificial Intelligence 8 (2025): 1458707.

[7] Bushra Alhijawi, Rawan Jarrar, Aseel AbuAlRub, and Arwa Bader. "Deep learning detection method for large language models-generated scientific content." Neural Computing and Applications 37, no. 1 (2025): 91-104.

[8] "Kaggle", [Online] Available: https://www.kaggle.com/, Last Accessed on: 26 April 2025.

[9] "DAIGT V2 Train Dataset", [Online] Available: https://www.kaggle.com/datasets/thedrcat/daigt-v2-train-dataset/data, Last Accessed on: 02 May 2025.

[10] "ChatGPT", [Online] Available: https://chatgpt.com/, Last Accessed on: 02 May 2025.

[11] "sentence-transformers/all-MiniLM-L6-v2", [Online] Available: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2, Last Accessed on: 06 May 2025.

[12] "sentence-transformers/all-mpnet-base-v2", [Online] Available: https://huggingface.co/sentence-transformers/all-mpnet-base-v2, Last Accessed on: 06 May 2025.

[13] "google-bert/bert-base-uncased", [Online] Available: https://huggingface.co/google-bert/bert-base-uncased, Last Accessed on: 06 May 2025.

[14] "MdSiamAnsary/ClassificationTasks", [Online] Available: https://github.com/MdSiamAnsary/ClassificationTasks, Last Accessed on: 13 June 2025.